

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-208307

(43)Date of publication of application : 25.07.2003

(51)Int.CI.

G06F 9/44

(21)Application number : 2002-007961

(71)Applicant : HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 16.01.2002

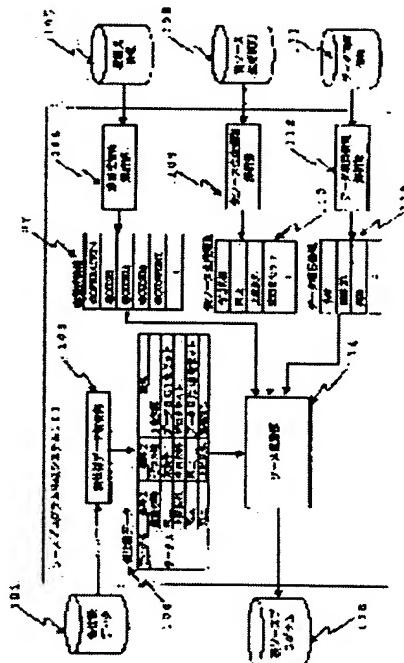
(72)Inventor : NISHIGUCHI MASATO

(54) SOURCE PROGRAM GENERATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a source program generation system capable of generating a source program with a complicated control structure from a program specification in a table form.

SOLUTION: The source program generation system for generating a corresponding source program from the program specification in the table form regarding a processing to data to be processed is provided with a means for reading the program specification in the table form which is constituted of a plurality of columns of conditional parts and processing parts and a source program generation means for extracting a Japanese sentence for branching control described in the read program specification and for generating a source program with branching structure corresponding to the Japanese sentence for branching control based on the executed Japanese sentence for branching control and preset source generation rules.



LEGAL STATUS

[Date of request for examination] 15.07.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-208307

(P2003-208307A)

(43)公開日 平成15年7月25日(2003.7.25)

(51)Int.Cl.⁷

G 0 6 F 9/44

識別記号

F I

G 0 6 F 9/06

テ-マコト[®](参考)

6 2 0 A 5 B 0 7 6

審査請求 未請求 請求項の数 3 O L (全 10 頁)

(21)出願番号

特願2002-7961(P2002-7961)

(22)出願日

平成14年1月16日(2002.1.16)

(71)出願人 000233055

日立ソフトウェアエンジニアリング株式会社

神奈川県横浜市鶴見区末広町一丁目1番43

(72)発明者 西口 真人

神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会社内

(74)代理人 100088720

弁理士 小川 真一

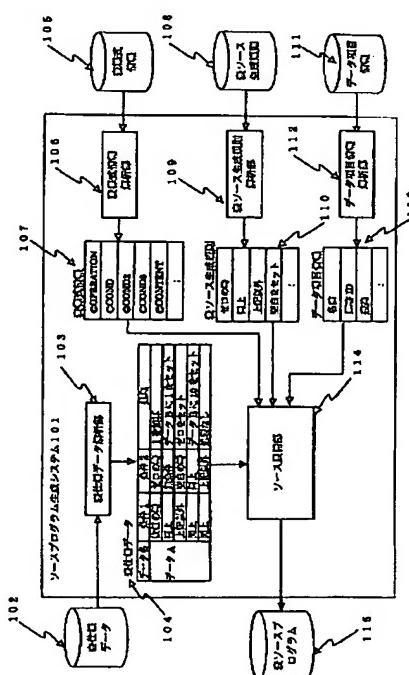
F ターム(参考) 5B076 DA03 DD04

(54)【発明の名称】 ソースプログラム生成システム

(57)【要約】

【課題】 表形式のプログラム仕様書から複雑な制御構造のソースプログラムを生成することができるソースプログラム生成システムを提供する。

【解決手段】 処理対象のデータに対する処理に関する表形式のプログラム仕様書から対応するソースプログラムを生成するソースプログラム生成システムであって、複数列の条件部と処理部で構成された表形式のプログラム仕様書を読み込む手段と、読み込んだプログラム仕様書に記述された分岐制御用の日本語文を抽出し、その抽出した分岐制御用の日本語文と予め設定されたソース生成規則に基づき、前記分岐制御用の日本語文に対応する分岐構造のソースプログラムを生成するソースプログラム生成手段とを備えることを特徴とする。



【特許請求の範囲】

【請求項1】 処理対象のデータに対する処理に関する表形式のプログラム仕様書から対応するソースプログラムを生成するソースプログラム生成システムであって、複数列の条件部と処理部で構成された表形式のプログラム仕様書を読み込む手段と、

読み込んだプログラム仕様書に記述された分岐制御用の日本語文を抽出し、その抽出した分岐制御用の日本語文と予め設定されたソース生成規則に基づき、前記分岐制御用の日本語文に対応する分岐構造のソースプログラムを生成するソースプログラム生成手段とを備えることを特徴とするソースプログラム生成システム。

【請求項2】 前記の分岐制御用の日本語文として表形式のプログラム仕様書における記述位置によって定まる同じ分岐に属することを指定するものと、他の分岐に属することを指定するものがあり、前記ソースプログラム生成手段はこれらの日本語文を参照して分岐のネスト構造を判断することを特徴とする請求項1に記載のソースプログラム生成システム。

【請求項3】 前記ソースプログラム生成手段がプログラム仕様書における分岐のネスト構造からソースプログラムを生成する際に、プログラムの制御構造に応じてソースプログラムの生成規則を定義した表仕様情報を備えることを特徴とする請求項1または2に記載のソースプログラム生成システム。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、業務プログラムの開発において、プログラム仕様を表形式の仕様書を用いて定義し、その定義情報からソースプログラムを生成するソース生成システムに関する。

【0002】

【従来の技術】 従来、特開平3-286226号公報(発明の名称: プログラム生成方法及びシステム)や、特開平5-265725号公報(発明の名称: ソースプログラム生成システム)に開示されているように、表形式やマトリクス形式の仕様書の枠に定義されている日本語文を入力することで、定義可能なソース生成規則を用いて、ソースプログラムを生成するシステムがある。また、特開平7-98650号公報(発明の名称: ソースプログラム生成方法およびシステム)に開示されているように、単純な条件を表現する日本語文を箇条書きにするだけで複雑な条件式のソースプログラムを生成するシステムがある。

【0003】 これらのシステムの特徴は、日本語文に対応するソースコードを定義することができることと、仕様書を条件部と動作部に分割することによって制御構造をソース生成規則で定義していることと、仕様書の枠に入力される日本語文に対するソースコード及び制御構造のソース生成規則を用いて変換処理を繰り返すことによ

ってソースプログラムを生成し、マトリクス仕様書や表仕様書に定義された条件から選択文や反復文を用いた制御構造のソースプログラムを生成することにある。

【0004】

【発明が解決しようとする課題】 しかし、上記の従来の技術で用いられているマトリクス仕様書からのソース生成システムは、制御を単純な判定で分岐させることはできるが、表仕様書からのソース生成は、条件部に定義した条件から連続したIF文を生成するだけである。すなわち、条件式の真偽によって何段階にも判定を分岐させるような複雑な制御構造のソースプログラムを生成することはできない。このため、実際のソフトウェア開発で要求される仕様書を簡単に定義することが困難であるという問題がある。

【0005】 本発明の目的は、表形式のプログラム仕様書から複雑な制御構造のソースプログラムを生成することができるソースプログラム生成システムを提供することにある。

【0006】

【課題を解決するための手段】 上記目的を達成するため、本発明は、処理対象のデータに対する処理に関する表形式のプログラム仕様書から対応するソースプログラムを生成するソースプログラム生成システムであって、複数列の条件部と処理部で構成された表形式のプログラム仕様書を読み込む手段と、読み込んだプログラム仕様書に記述された分岐制御用の日本語文を抽出し、その抽出した分岐制御用の日本語文と予め設定されたソース生成規則に基づき、前記分岐制御用の日本語文に対応する分岐構造のソースプログラムを生成するソースプログラム生成手段とを備えることを特徴とする。また、前記の分岐制御用の日本語文として表形式のプログラム仕様書における記述位置によって定まる同じ分岐に属することを指定するものと、他の分岐に属することを指定するものがあり、前記ソースプログラム生成手段はこれらの日本語文を参照して分岐のネスト構造を判断することを特徴とする。さらに、前記ソースプログラム生成手段がプログラム仕様書における分岐のネスト構造からソースプログラムを生成する際に、プログラムの制御構造に応じてソースプログラムの生成規則を定義した表仕様情報を備えることを特徴とする。

【0007】

【発明の実施形態】 以下、本発明を図面に示す実施形態に基づいて詳細に説明する。図1は、本発明を適用したソースプログラム生成システムの実施形態を示すブロック図である。図1において、本発明に係るソースプログラム生成システム101は、あるデータに対する操作の仕様として、複数列の条件部と処理部で構成されている表仕様書が格納されている表仕様データ102を解析し、メモリ中に表仕様データ103として書込む表仕様データ解析部103を備えている。また、条件部及び処

理部の列に関する生成規則が定義された表様式情報105を解析し、各列の関連と生成規則をメモリ中に表様式情報107として書込む表様式情報解析部106を備えている。

【0008】また、表仕様書で使用される日本語文に対応するソースコードが定義された表ソース生成規則108の情報を解析し、メモリ中に表ソース生成規則110として書込む表ソース生成規則解析部109を備えている。また、表仕様書の処理対象であるデータ項目やまた表仕様書上で参照される他のデータ項目の情報が格納されているデータ項目情報111を解析し、メモリ中にデータ項目情報113として書込むデータ項目情報解析部112を備えている。また、表仕様データ104と表様式情報107と表ソース生成規則110とデータ項目情報113を用いて分岐構造を解析しながら最終的なソースプログラムである表ソースプログラム115を生成するソース展開部114を備えている。

【0009】図2は、表様式情報解析部106によって解析された表様式情報107の例である。ここで示す表様式情報107は、生成規則の名前を示す枠記号201と、生成規則の展開を示す枠定義202から構成される。ここで、図2の例で示されている5つの枠記号について説明する。まず、符号203で示す「@OPERATION」は、表様式情報107の先頭を示す枠開始記号であり、その枠定義は生成規則が「@COND」から展開されることを示している。次の符号204～206で示す「@COND」、「@COND2」、「@COND3」までの枠定義は、条件部の生成規則を定義している。図2では3列の条件部を持つ表様式情報を示しているが、目的に応じて「@COND4」や「@COND4」を定義することによって、任意列数の条件部を持つ表仕様書を利用することが可能である。次の符号の214で示す「@CONTENT」は処理部の生成規則を定義している。図2では、条件部以外には処理部を1つだけ（「@CONTENT」）を持つ表様式情報を示しているが、コメント欄用や特殊な制御命令などを定義する列のために任意の名前の枠記号を追加することが可能である。

【0010】次に、条件部の枠定義中の記号について説明する。図2の表様式情報107の条件部は、3段階にネストされるIF文のソースコードを生成するためのものであり、それぞれの枠記号が1つのIF文の生成規則に対応している。符号207で示す「&:COND」や符号208の「&:COND2」、符号209の「&:COND3」、符号210の「&:CONTENT」は、表仕様書に入力される日本語文に対応する置換記号であり、ソース展開部114によって日本語文から展開されたソースコードと置き換えられる。符号211の「@COND2」や符号212の「@COND3」、符号213の「@CONTENT」のような枠定義中に使用されている枠記号は、次の展開記号を示している。すなわち、符号204の「@COND」のTHEN句の処理文は、符号205の「@COND2」の展開結果であり、符号205

の「@COND2」のTHEN句の処理文は、符号206の「@COND3」の展開結果であり、符号206の「@COND3」のTHEN句の処理文は、符号214の「@CONTENT」の展開結果であることを示している。

【0011】一方、符号215の「@COND2#EXCEPTION」や符号216の「@COND3#EXCEPTION」、符号217の「@CONTENT#EXCEPTION」のように、枠記号の後ろに「#EXCEPT」が付加されているものは、ELSE句用の展開記号を示している。すなわち、ソース展開部114は分岐構造を解析することによって、THEN句として展開するか、またはELSE句として展開するかを判断し、いずれかの枠記号の位置にソースコードを展開する。このように枠定義202に次の展開先を示す枠記号201をTHEN句用とELSE句用に分類して記述することで表仕様書に入力された情報の分岐構造に応じた生成規則を定義することができる。

【0012】図3は、表ソース生成規則解析部109によって解析された表ソース生成規則110の例である。表ソース生成規則110は、表仕様書上で表現される日本語文301と、日本語文301に対応して表ソースプログラム上で表現されるソースコード302から構成される。符号303で示す「同上」と符号305で示す「上記以外」は、分岐制御用の日本語文であり、これらの日本語文に対応するソースコードの「#SAME」304と「#EXCEPT」306は、ソース展開部114が分岐構造を解析する際に検出する特殊な文字列である。また、符号307で示す「無条件」も特殊な日本語文であり、この日本語文に対応するソースコードの「#NOSTATEMENT」308は、条件部に条件がないこと示している。図3に示すその他の日本語文「ゼロの時」、「数値の時」などは、表仕様書の条件部や処理部で使用するためのものであり、ユーザは日本語文とソースコードを自由に追加・変更することができる。なお、日本語文中の「□」には、表仕様書を定義する際に任意のデータ項目を指定することができる。ソース展開時には、「□」に指定されたデータ項目は、対をなすソースコードの「\$1」や「\$2」に対応して、データ項目の英語名に置き換えられる。これは表ソース生成規則110の再利用性を向上させるためのものである。また、ソースコード中の「&:ITEM」は、処理対象のデータ項目の英語名に置き換えられる。

【0013】図4は、データ項目情報解析部112によって解析されたデータ項目情報113の例である。データ項目情報113は、データ項目の表仕様書上の表現である日本語名401と、ソースプログラム上の表現である英語名402から主に構成される。

【0014】図5は、表仕様データ解析部103によって解析された表仕様データ104の例である。この例は、2つの条件部と処理部を持つ表様式の場合のものである。図6及び図7と図8は、ソース展開部114の処

理の流れを表わすフローチャートである。ソース展開部114の処理は、表仕様データ104を参照して分岐構造を解析する判定処理と、表様式情報107を参照してソースコードを展開する処理に分類される。図6と図7は、分岐構造を解析する判定処理のフローチャートであり、図8はソース展開処理のフローチャートである。

【0015】以下、それぞれのフローチャートについて詳細に説明する。分岐構造を解析する判定処理は、分岐のネスト構造を解析するという性質から、処理の流れは、再帰的なものになっている。図6と図7では、この再帰呼び出しの処理を「枠記号展開ルーチン呼び出し」と表現している。図6のフローチャートは、その再帰的な処理の初回の呼び出しを示したもので、呼び出しに必要な情報を取得及び設定している。まず、ステップ601の解析開始枠記号の取得処理では、表様式情報107の中から203「@OPERATION」を検索し、表様式情報107の先頭を示す枠記号を取得する。この枠記号は、再帰的な処理の初回に解析される枠記号であり、枠記号展開ルーチン呼び出しのパラメータに指定される。次のステップ602の枠記号展開ルーチン呼び出し処理は、前述のように再帰的な処理の呼び出しである。パラメータには、ステップ601の解析開始枠記号の取得処理で得た枠記号と、表仕様データ104の行の中で解析を行う範囲を示すための開始位置と終了位置を指定する。

【0016】初回の呼び出しでは、全ての行を解析の範囲とするため、1行目と最終行を指定している。このように初回に解析する枠記号と、解析の範囲を表仕様データ104の全範囲とすることによって、後述する再帰的なソース展開処理によって、最終的には、表仕様データ104全体の分岐構造を分析した結果得られるプログラムコードを取得することができる。

【0017】図7のフローチャートは、再帰的な分岐構造解析とソース展開を行う枠記号展開ルーチンの処理を示している。この処理のアルゴリズムの特徴は、図5で示したような表仕様データ104の各列の日本語文が、分岐構造におけるTHEN側とELSE側のどちらに分類されるかを再帰的に判定して、それに応じてソースを展開していることである。ソースの展開には、図2に符号207で示した「&:COND」や符号208で示した「&:COND2」などを置き換える枠の置換記号に関する処理と、符号211で示した「@COND2」や符号212で示した「@COND3」などの枠記号の展開に関する2種類がある。分類の判定は、図3に示した分岐制御用の日本語文301を用いて行われる。

【0018】次に、処理の詳細について説明する。まず、ステップ701における初期設定処理では、範囲基準位置702と分岐フラグ703を初期化している。範囲基準位置702は、1つのTHENまたはELSEの範囲を特定する際に、一時的に記録しておく範囲の開始位置である。一方、分岐フラグ703は、現在範囲を特定してい

るのが、THENまたはELSEのどちらであるかを示している。次のステップ704における枠の置換記号の展開処理では、表仕様データ104でパラメータの枠記号と開始位置が示す欄のソースコードを表ソース生成規則110とデータ項目情報113を用いて生成し、図2の符号207で示した「&:COND」や符号208で示した「&:COND2」など枠置換記号とソースコードを置き換える。

【0019】続くステップ705以降の繰り返し処理では、現在位置がパラメータで指定されている開始位置から終了位置までの間、分岐構造の判定処理を各行について繰り返す。まず、ステップ706において現在位置の日本語文が「無条件」、「同上」、「上記以外」、「その他」のいずれであるかを判定し、現在位置の日本語文に対応するソースコード302を表ソース生成規則110から参照することで、分岐制御用のソースコード「#SAME」304、「#EXCEPT」306、「#NOSTATEMENT」308、及びその他の日本語文に対応するソースコードのいずれに該当するかを判定する。そして、その判定結果に応じて、「無条件」707と「同上」708、「上記以外」709、「その他」710の4種類に処理の流れを振り分けている。日本語文が「無条件」に該当する場合には、ステップ711において次の枠記号の展開処理を行い、さらにステップ712で範囲基準位置の設定を行う。

【0020】ステップ711の次の枠記号の展開及びステップ713における次の枠記号の展開とステップ716における次の枠記号の展開、ステップ718における次の枠記号の展開は全く同じ処理である。この処理は、判定処理によって1つのTHENまたはELSEの範囲が確定した時に、同じ範囲で次にネストされている枠記号について枠記号展開ルーチンからソースコードを取得し、THEN側またはELSE側として展開する。これらの詳細は図9を用いて説明する。

【0021】ステップ712における範囲基準位置の設定処理では、次のTHENまたはELSEの範囲の開始位置を再設定している。したがって、ステップ706における現在位置の日本語文が「無条件」707の場合には、強制的に次の枠記号の展開を行う。次に、現在位置の日本語文が「同上」708の場合には、何も行わない。すなわち、範囲の特定を継続する。

【0022】一方、現在位置の日本語文が「上記以外」709の場合には、ステップ713において次の枠記号の展開を行った後に、ステップ714で次の範囲の開始位置再設定と分岐フラグをELSEに設定している。すなわち、「上記以外」709が現れるまでは、分岐フラグはTHENでありTHEN側の範囲であるが、「上記以外」709が現れることによって、現在位置までをTHEN側と特定し、それ以降をELSE側として解析を開始する。

【0023】また、現在位置の日本語文が「その他」710の場合には、ステップ715において「現在位置

範囲基準位置」かを判定し、現在位置が範囲基準位置と異なる場合のみステップ716で次の枠記号の展開とステップ717次の範囲の開始位置再設定と分岐フラグをTHENに設定している。すなわち、枠記号展開ルーチン呼び出しが呼び出された直後の1回目の繰り返し時のみ、現在位置が範囲基準位置と等しくなるので、この時以外は制御用の日本語文以外の場合には、強制的に次の枠記号の展開を行う。

【0024】このように4種類の判定によって表仕様データ104の開始位置から終了位置までの間でのTHENとELSEの切り替え位置を検出することができる。また、ステップ718における次の枠記号の展開処理では、ステップ705の繰り返し処理が終了することによって確定する範囲のためのものである。

【0025】図8のフローチャートは、図7のフローチャートの説明で簡略的に説明した次の枠記号の展開を示している。この処理は、図7で説明した判定処理などによって確定したTHENまたはELSEの範囲で、現在の枠からネストされている次の枠記号について再帰的に展開する。展開によって得られたその範囲のソースコードは、分岐フラグによって展開位置をTHENの位置またはELSEの位置に展開する。まず、ステップ801における範囲のチェック処理では、範囲基準位置が現在位置より小さいかをチェックすることで、開始位置が無条件である場合などで範囲異常になることを防止している。

【0026】続くステップ802の枠記号展開ルーチン呼び出し処理では、図7の枠記号展開ルーチンを呼び出す。その時のパラメータは、表様式情報107の現在の枠記号に対応する枠定義の中から検出できる他の枠記号、確定した範囲の開始位置を示す範囲基準位置と、範囲を確定した時の1つ前の位置である。呼び出された枠記号展開ルーチンからは、パラメータで指定された枠記号の列において指定の範囲のソースコードを取得できる。続くステップ803では、枠記号展開ルーチンから得られたソースコードの展開位置を分岐フラグによって判定している。分岐フラグがTHENの場合には、ステップ804の「次の枠記号」位置に展開によってTHENの位置、すなわち図2の符号211で示した「@COND2」や符号212の「@COND3」、符号213の「@CONTENT」などの位置にソースコードを展開する。

【0027】一方、分岐フラグがELSEの場合には、ステップ805の「ELSE句用次の枠記号」位置に展開する処理によってELSEの位置、すなわち枠記号に「#EXCEPTION」を付加したELSE句用の展開記号である「@COND2#EXCEPTION」215や「@COND3#EXCEPTION」212の位置にソースコードを展開する。

【0028】図9(a)、(b)は、図5の表仕様データ104をソース展開部114によって処理した結果を示している。901は、分岐構造の解析処理によって、図5の表仕様データ104の各欄が、どのような順番で

THENまたはELSEのどちらに判定されたかを矢印とその中の番号によって示している。符号902で示すTHENの範囲や符号903で示すELSEの範囲で補足しているよう、白い矢印は、THEN側として判定され、黒色の矢印はELSE側として判定されている。このように、「同上」がある間は範囲が継続していると判定され、上記以外があると範囲が反転するという簡単な判定処理によって、条件分岐のネスト構造を分析することができる。

【0029】図9(b)に示す904は、ソース展開部114が図9のように判定した表仕様データから最終的なソースコードを生成した結果であり、THENとELSEによる複雑なネスト構造を持つIF文を生成していることが分かる。また、ソースコードの左側の矢印は、ソースコードの各部分と表の矢印の対応を示しており、白い矢印がTHEN側として判定された部分のソースコード、黒色の矢印がELSE側として判定された部分のソースコードを示し、矢印内の丸印数字は図9(a)の数字と対応している。

【0030】

【発明の効果】以上の説明から明らかなように、本発明によれば、表形式の仕様書に分岐制御用の日本語文を含め、かつ分岐用のソース生成のルールを定義することによって、分かりやすい表形式のフォーマットで複雑な分岐構造の仕様書から対応するソースコードを生成することが可能になる。

【図面の簡単な説明】

【図1】本発明の一実施形態の概略構成を示すシステム構成図である。

【図2】表様式情報の例を示す説明図である。

【図3】表ソース生成規則の例を示す説明図である。

【図4】データ項目情報の例を示す説明図である。

【図5】表仕様データの例を示す説明図である。

【図6】ソース展開処理における初回のサブルーチン呼び出しの詳細を示すフローチャートである。

【図7】ソース展開処理の詳細を示すフローチャートである

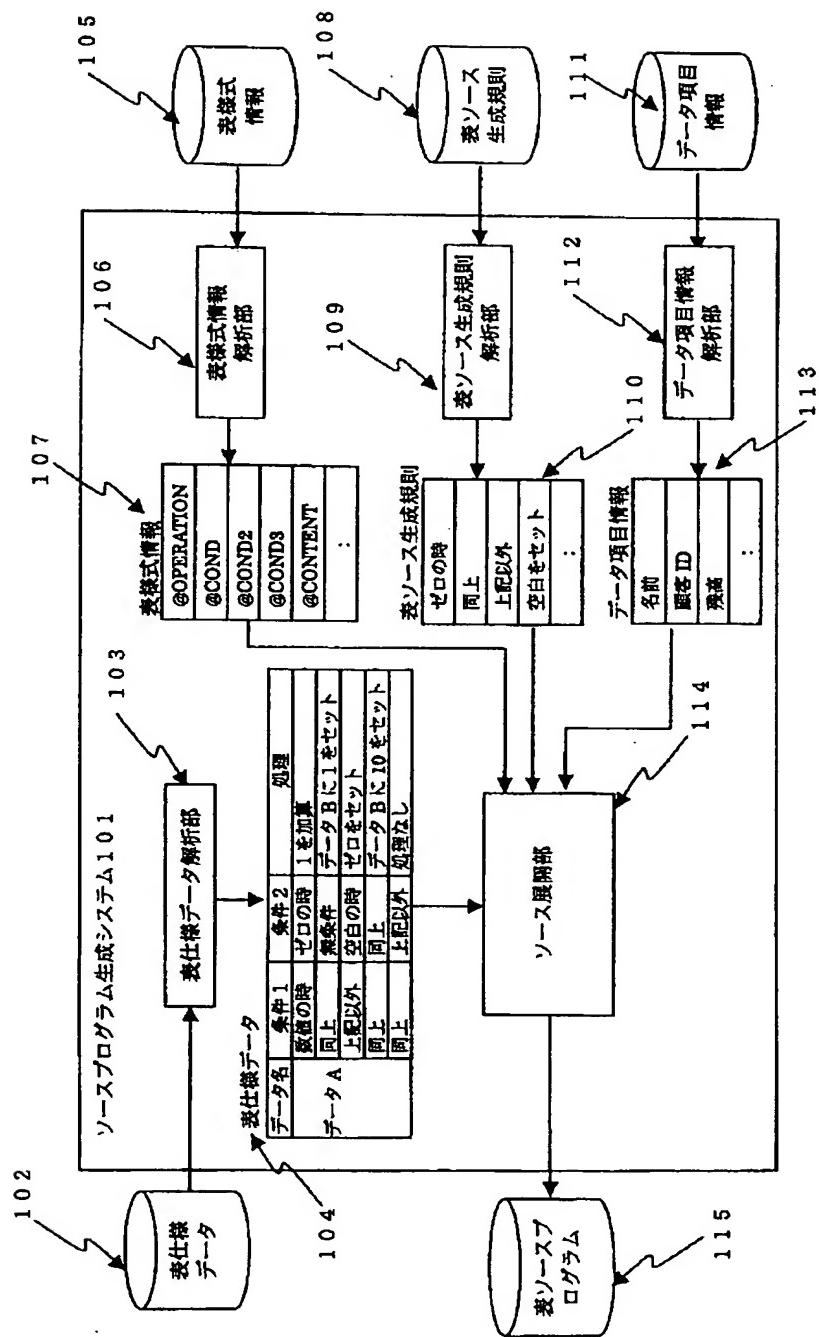
【図8】ソース展開処理で次の枠記号の展開を行う処理の詳細を示すフローチャートである

【図9】図5の表仕様データの場合にソース展開処理で判定される分岐のネスト構造の範囲と生成されるソースコードの例を示す説明図である。

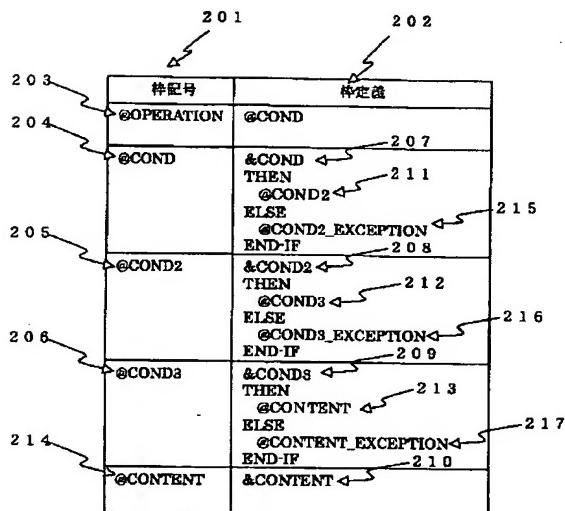
【符号の説明】

101…ソースプログラム生成システム、102…表仕様データ、103…表仕様データ解析部、104…表仕様データ、105…表様式情報、106…表様式情報解析部、107…表様式情報、108…表ソース生成規則、109…表ソース生成規則解析部、110…表ソース生成規則、111…データ項目情報、112…データ項目情報解析部、113…データ項目情報。

【図1】



【図2】



【図3】

301	日本語文	ソースコード
302		
303	#SAME	304
305	#EXCEPT	306
307	#NOSTATEMENT	308
309	IF &ITEM = ZERO	
310	IF &ITEM IS NUMERIC	
311	IF &ITEM = \$1	
312	IF \$1 = \$2	
313	MOVE SPACE TO &ITEM	
314	MOVE ZERO TO &ITEM	
315	MOVE \$1 TO &ITEM	
316	COMPUTE &ITEM = &ITEM + 1	
317	COMPUTE &ITEM = &ITEM - 1	
318	CONTINUE	

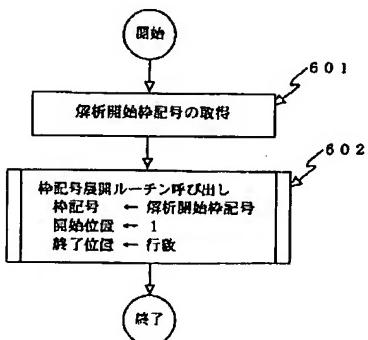
【図4】

401	日本語名	英語名
402		
顧客ID	CUSTID	
口座番号	ACCOUNT	
残高	BALANCE	
取引額	TRADE	

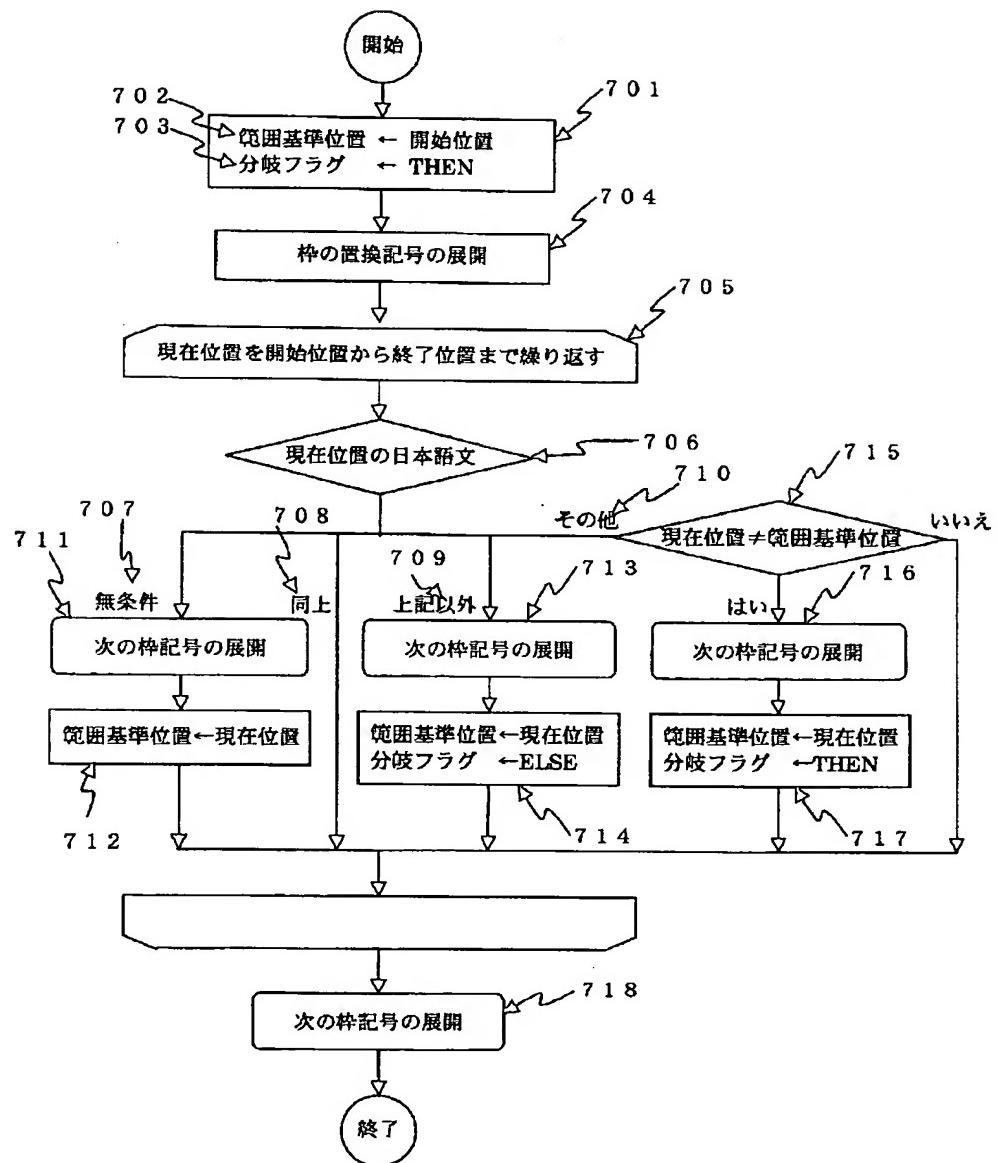
【図5】

データ名	条件1	条件2	処理
データA	数値の時	ゼロの時	1を加算
	同上	無条件	データBに1をセット
	上記以外	空白の時	ゼロをセット
	同上	同上	データBに0をセット
	同上	上記以外	処理なし

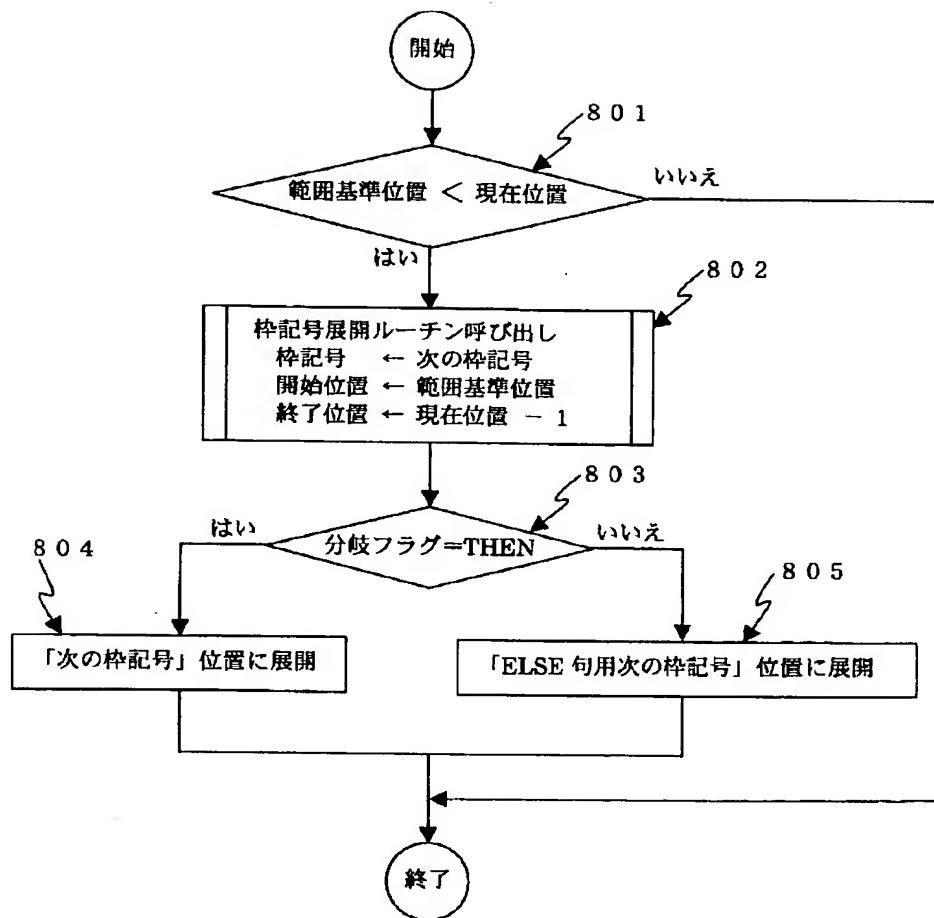
【図6】



【図7】



【図8】



【図9】

